

**AMENDMENTS TO THE CLAIMS**

1. (Original) A method for mediating event records between a generation layer of events and an operation system layer of events in a communications network by means of a mediation layer of events, which includes at least one first self-contained component of the mediation layer and at least one second self-contained component of the mediation layer, which operates independently of each first component of the mediation layer, and at least one buffer, the method comprising
  - collecting event records from an element of the generation layer of events substantially continuously as a stream, by the at least one first self-contained component of the mediation layer,
  - processing the collected event records substantially continuously, wherein the step of processing includes:
    - writing the output from each of the at least one first self-contained component into one of said at least one buffer, and
    - reading the input for each of the at least one second self-contained component from one of said at least one buffer,
  - delivering the processed event records to an element of the operation system layer of events substantially continuously as a stream, by the at least one second self-contained component of the mediation layer.
2. (Original) A method according to claim 1, wherein at least part of the step of processing event records is performed by at least one first self-contained component of the mediation layer.

3. (Currently amended) A method according to claim 1 ~~or 2~~, wherein at least part of the step of processing event records is performed by at least one second self-contained component of the mediation layer.
4. (Currently amended) A method according to ~~any of claims 1 to 3~~ claim 1, wherein at least part of the step of processing event records is performed by at least one third self-contained component of the mediation layer that operates independently of the other self-contained components of the mediation layer.
5. (Currently amended) A method according to ~~any of claims 1 to 4~~ claim 1, wherein at least two different hosts are used such that at least one of the self-contained components of the mediation layer runs in a first host and at least one of the other self-contained components runs in another host.
6. (Original) A method according to claim 4, comprising the steps of
  - delivering event records from each of the first self-contained components of the mediation layer to the at least one third self-contained component of the mediation layer via at least one buffer, and
  - delivering event records from the third self-contained components of the mediation layer to one of the at least one second self-contained component of the mediation layer via at least one buffer.
7. (Currently amended) A method according to ~~any of claims 1 to 6~~ claim 1, wherein the event records are passed through at least three self-contained components of the mediation layer,

starting from one of the first self-contained components, then through at least one third self-contained component and finally through one of the second self-contained components.

8. (Original) A method according to claim 7, wherein the step of delivering event records comprises

- writing the event records output by a preceding self-contained component of the mediation layer into a buffer, and
- reading the buffer substantially continuously by the subsequent self-contained component of the mediation layer.

9. (Original) A method according to claim 8, wherein the preceding self-contained component of the mediation layer outputs event records into the buffer one by one, and the subsequent self-contained component of the mediation layer reads event records from the buffer one by one.

10. (Original) A method according to claim 8, wherein the preceding self-contained component of the mediation layer outputs event records into the buffer grouped into small groups of event records, and the subsequent self-contained component of the mediation layer reads event records from the buffer in small groups of event records.

11. (Currently amended) A method according to ~~any of claims 8 to 10~~ claim 8, wherein at least two separate self-contained components of the mediation layer write event records into the same one buffer.

12. (Currently amended) A method according to ~~any of claims 8 to 11~~ claim 8, wherein at least two separate self-contained components of the mediation layer read event records from one and the same buffer.

13. (Currently amended) A method according to ~~any of claims 8 to 12~~ claim 8, wherein after reading an event record from a buffer, a copy of the event record is retained in the buffer, and removed from the buffer only after successfully outputting the event record from the reading self-contained component of the mediation layer.

14. (Original) A method according to claim 13, wherein the retained event record is mark with status information indicating the “under processing” status of the event record.

15. (Currently amended) A method according to ~~any of claims 1—14~~ claim 1, comprising the steps of monitoring by a monitoring system the operation of the self-contained components of the mediation layer and, in case of failure of any of the self-contained components, automatically setting up a new self-contained component to replace the failed component.

16. (Currently amended) A method according to ~~any of claims 1—15~~ claim 1, comprising the steps of monitoring by a monitoring system the production capacity of the self-contained components of the mediation layer and, in case of insufficient production capacity of any of the self-contained components, automatically setting up an auxiliary self-contained component parallel to the self-contained component with insufficient production capacity.

17. (Currently amended) A method according to ~~any of claims 1—16~~ claim 1, wherein the auxiliary self-contained component is set up to run in a host different to the host in which the self-contained component with insufficient production capacity runs.

18. (Currently amended) A method according to ~~any of claims 1—16~~ claim 1, comprising the steps of

- receiving event records from the step of collecting in a source system format,
- converting the received event records into a mediation layer format,
- supplying the collected event records to the step of processing in the mediation layer format,
- receiving the processed event records from the step of processing in the mediation layer format,
- converting the processed event records into an operation system layer format, and
- supplying the processed event records to the step of delivering in the operation system layer format.

19. (Currently amended) A method according to ~~any of claims 1—18~~ claim 1, wherein the step of processing event records comprises at least one of the following: validating and analysing event records, enrichment of event records, aggregation and correlation of event records, formatting of event records and rating.

20. (Currently amended) A method according to ~~any of claims 1—19~~ claim 1, wherein each of the self-contained components operates independently and continuously once started.

21. (Currently amended) A method according to ~~any of claims 1 to 20~~ claim 1, comprising the steps of

- stopping the operation of a self-contained component by the self-contained component itself, and
- performing said step of stopping the operation by the self-contained component only if instructed so by a manager component of the mediation layer.

22. (Currently amended) A method according to ~~any of claims 1—21~~ claim 1, comprising the steps of

- providing each of the self-contained components with its own individual settings, and
- each of the self-contained components functioning according to its own individual settings.

23. (Original) A method according to claim 22, wherein said individual settings of each of the self-contained components include

- a node base part of the settings, which is identical to the node base parts of the other self-contained components within the mediation layer, and
- a node application part of the settings, which contain custom processing rules and which is different to the node application parts of at least most of the other self-contained components within the mediation layer.

24. (Original) A system for handling event records in a communications network between a generation layer of events and an operation system layer of events, the system comprising:

- at least two independent node components for processing event records, each of the independent node components having its own settings according to which the node operates independently of other components of the system, at least two of the independent node components being configured to handle event records in series such that the first independent node component outputs into a buffer and the second independent node component reads its input from the buffer,
- at least one node manager component for configuring the node components, starting up the node components, monitoring the functioning of the node components and stopping the node components, when required, and
- a system database for managing all configuration information of each component and for storing information on handled events.

25. (Original) A system according to claim 24, wherein more than one independent node component have been configured to output into the same buffer.

26. (Currently amended) A system according to claim 24 ~~or 25~~, wherein more than one independent node component have been configured to read its input from the same buffer.

27. (Currently amended) A system according to ~~any of claims 24 to 26~~ claim 24, wherein at least two of the independent node components have been configured to input, process and output event records substantially continuously.

28. (Currently amended) A system according to ~~any of claims 24 to 27~~ claim 24, comprising a user interface for controlling, monitoring and configuring the system.

29. (Currently amended) A system according to ~~any of claims 24 to 28~~ claim 24, wherein the configuration or settings of any component can be changed by a supervisor at any time, without stopping the handling process.

30. (Currently amended) A system according to ~~any of claims 24 to 29~~ claim 24, wherein the tasks undertaken by the node components include collecting events records from a communication network, aggregating event records, converting event records, analysing event records, correlating event records, enriching event records, formatting event records, rating events and/or delivering event records.

31. (Currently amended) A system according to ~~any of claims 24 to 30~~ claim 24, which is configured to process event records in several, simultaneously operating, and at least partly parallel streams.

32. (Currently amended) A system according to ~~any of claims 24 to 31~~ claim 24, comprising at least two audit trail counters for counting auditing values, such as number of: incoming records, rejected records, reprocessed records, records residing in a specific node component, records omitted due to filtering, records expired or deleted, new records created due to splitting or duplication, new records generated that are not related to input records, input records sent to aggregation/correlation process, records that were merged due to aggregation or correlation, resulting records that were completed and came out from the aggregation/correlation process, resulting records that were flushed out from the aggregation/correlation process, records left to a specific node component and/or records written out.



33. (Currently amended) A system according to ~~any of claims 24 to 32~~ claim 24, comprising at least one audit trail function for checking that no data is lost within the system.

34. (Currently amended) A system according to ~~any of claims 24 to 33~~ claim 24, comprising at least one data storage component, wherein at least one node component is configured to write information on all of the events processed by the node component.

35. (Currently amended) A system according to ~~any of claims 24 to 34~~ claim 24, wherein the node manager component is configured to start up a new node component in case a node component in the system fails such that the new node component replaces the function of the failed component in the processing chain.

36. (Currently amended) A system according to ~~any of claims 24 to 35~~ claim 24, wherein the node manager component is configured to start up a new node component parallel to a functioning node component in case the processing capacity of the system has to be raised.

37. (Currently amended) A system according to ~~any of claims 24 to 36~~ claim 24, wherein each of the node components comprise a node base providing basic functionality of the node component and an application containing processing rules, according to which the node component processes the event records input to the node component.

38. (Original) A system according to claim 37, wherein the node bases of the node components are identical to each other.

39. (Currently amended) A system according to claim 37 ~~or 38~~, wherein the node base includes an input module, an output module, an API module, a configuration module and an audit module.

40. (Currently amended) A system according to ~~any of claims 24 to 39~~ claim 24, wherein the node components have been configured to continue their independent operation until instructed otherwise by the node manager component.

41. (Currently amended) A system according to ~~any of claims 24 to 40~~ claim 24, comprising at least two separate hosts, each of the hosts running at least one of the independent node components.

42. (Original) A computer program product for a system for handling event records in a communications network between a generation layer of events and an operation system layer of events, which system comprises independent nodes for processing event records, the computer program product comprising:

- a node base program means capable of providing basic software functionality for an independent node, said basic software functionality including an external interface of the node and an internal interface of the node,
- an application programming interface means for receiving application programs for independent nodes, which application programs are capable of interfacing with the internal interfaces of the node components,

- a node manager program means for setting up at least one node manager that is capable of constructing, configuring, starting up, monitoring and stopping the independent nodes, and
- a user interface program means for setting up a user interface for configuring the at least one node manager.

43. (Original) A computer program product according to claim 42, wherein the node manager program means include program code means to direct a node manager to construct independent nodes by combining a copy of node base program means and an application program.

44. (Currently amended) A computer program product according to claim 42 ~~or 43~~, wherein the application program contains logical rules according to which the node processes the event records input to the node.

45. (Currently amended) A computer program product according to ~~any of claims 42 to 44~~ claim 42, wherein the external interface of the node enables the node to communicate with other nodes and the node manager.

46. (Currently amended) A computer program product according to ~~any of claims 42 to 45~~ claim 42, wherein the node manager program means include program code means to direct a node manager, in case a node in the system fails, to construct, configure and start up a new node that replaces the function of the failed node.

47. (Currently amended) A computer program product according to ~~any of claims 42 to 46~~ claim 42, wherein the node manager program means include program code means to direct a node

manager, in case of insufficient production capacity of any of the nodes, to construct, configure and start up a new node parallel to the node with insufficient production capacity.

48. (Currently amended) A computer program product according to ~~any of claims 42 to 47~~ claim 42, wherein the application programming interface means are capable of supporting several programming languages.

49. (Currently amended) A computer program product according to ~~any of claims 42 to 48~~ claim 42, which is capable of configuring the nodes to form processing chains of serially connected independent nodes, for processing the event records.

50. (Original) A computer program product according to claim 49, which is capable of configuring the nodes in the processing chains to transfer event records from the preceding node in the chain to the subsequent node in the chain by means of a buffer.

51. (Currently amended) A computer program product according to ~~any of claims 42 to 50~~ claim 42, which is capable of configuring the nodes to function continuously and independently until instructed otherwise by the node manager.

52. (Currently amended) A computer program product according to ~~any of claims 42 to 51~~ claim 42, which supports multi-host execution and is capable starting up nodes in different hosts, and configuring the nodes in different hosts to form processing chains for processing the event records.